

Evolving local search heuristics for the integrated berth allocation and quay crane assignment problem

Tamer El-boghdadly
School of Computing
University of Portsmouth
Portsmouth PO1 3HE, UK
Email: tamer.el-boghdadly@port.ac.uk

Mohamed Bader-El-Den
School of Computing
University of Portsmouth
Portsmouth PO1 3HE, UK
Email: mohamed.bader@port.ac.uk

Dylan Jones
School of Mathematics
University of Portsmouth
Portsmouth PO1 3HF, UK
Email: dylan.jones@port.ac.uk

Abstract—Water Transportation is the cheapest transportation mode, which allows the transfer of very large volumes of cargo between continents. One of the most important types of ships used to transfer goods are the Container Ships, therefore, containerized trade volume is rapidly increasing. This has opened a number of challenging combinatorial optimization problems in container terminals. This paper focuses on the integrated problem Berth Allocation and Quay Crane Assignment Problem (BQCAP), which occur while planning incoming vessels in container terminals. We provide a Genetic Programming (GP) approach to evolve effective and robust composite dispatching rules (CDRs) to solve the problem and present a comparative study with the current state-of-art optimal approaches. The Computational results disclose the effectiveness of the presented approach.

Keywords—Berth Allocation; Quay Crane Assignment; Container Terminal Operations; Genetic Programming; Composite dispatching rules; Optimization; Scheduling;

I. INTRODUCTION

Container terminals (CTs) play an important role in the international trading as it is the most extended mean of transport of goods around the world. Moreover, there are a number of major developments in the waterway and the container terminal industry, that will have a direct impact on shipping, ports and container ships traffic. In order to cope with this rapid increase in the number of containers, maritime container terminals have to be ready to face these challenges and improve its services and quality for incoming vessels. Therefore, Seaport competition has increased considerably. 2014 the level had reached 171 million twenty-foot equivalent units (TEU), [1]. Table I presents the world's 10 busiest container ports in the world in terms of TEUs handled.

The overall aim of this study is to improve the seaside port operations through the development of new intelligent algorithms for optimising the operations in container terminals. In particular, reducing the total cost of port stay times for all vessels through reducing the total service costs and total operation cost for the incoming ships. For the best of our knowledge, there are no previous studies used the genetic programming approach to solve the BQCAP, and there is only one paper [2] that use GP and Genetic Algorithm (GA) to solve Quay Crane Scheduling Problem, the results indicates that using GP performs better than GA both in terms of average fitness and the best fitness. Therefore, it would be interesting to solve the BQCAP using the GP approach.

TABLE I: Top ten busiest container ports in the world ranked as of 2014 (Million TEUs)

Rank	Port, Country	2013	2014	% change 2014-2013
1	Shanghai, China	36.62	35.29	-3.62
2	Singapore, Singapore	32.60	33.87	3.89
3	Shenzhen, China	23.28	24.04	3.27
4	Hong Kong, China	22.35	22.20	-0.68
5	Ningbo, China	17.35	19.45	12.10
6	Busan, South Korea	17.69	18.68	5.61
7	Guangzhou, China	15.31	16.61	8.50
8	Qingdao, China	15.52	16.58	6.83
9	Dubai, United Arab Emirates	13.64	15.2	11.43
10	Tianjin, China	13.00	14.06	8.15

The contribution in this paper could be summarised in the following points: (1) The paper presents a genetic programming based approach for evolving dispatching rules for the BQCAP problem. In addition to the high performance of the evolved DRs, the main advantage of the approach is the "self-adaptability" of the proposed methods; due to the fact that almost all container terminal ports have different characteristics, the performance of DR based schedules varies significantly from port to another and therefore, it is important to manually tune and select the best-performing DRs. The proposed GP approach is "self-adaptable" which automatically discovers/evolves high performing DR using different sets of variables based on what is available in each berth. (2) The paper also presents an independent *Scheduler* for the BQCAP that could be combined with any appropriate optimization method. (3) The paper provides an analysis of a wide range of DRs and the GP approach and compare the results with well known BQCAP benchmark. (4) We develop a new technique to improve quay spaces utilization while scheduling.

The paper is organized as follows. In Section III, we investigate the current state of the art for both the berth allocation and the quay crane assignment problem in the literature. The problem description are described in Section II. In Section II-B, we provide our model, notations, and mathematical model used to solve the BQCAP. The proposed solution design framework is explained in Section IV. Section V shows the experimental design and the results obtained by the proposed methods and a comparative study with the results of using different SPR and CDRs. Finally, Section VI concludes the paper and presents future research directions.

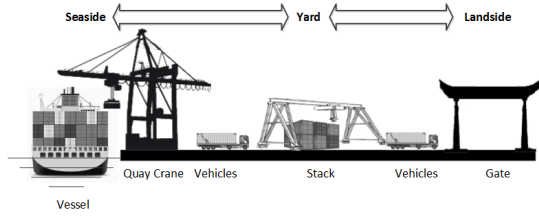


Fig. 1: Container terminal system for transshipment process

II. PROBLEM DESCRIPTION

This paper presents a genetic programming based method for the integrated berth allocation and quay crane assigning problem BQCAP. Before going through the mathematical definition, it is important to outline the main components and processes in container terminal. Therefore, this section start by proving and overview of the main operations in container terminals, followed by a definition of the BQCAP.

A. Container Terminals: systems and processes

A container terminal (CT) is a type of port that has the facility where container ships are transhipped. Container ship (CS) is a type of ship or vessel that carries containers to transfer goods from one place to another. Containers are large metal boxes of a standard design and size for the transport of goods; The most commonly used sizes today are the twenty-foot-equivalent Unit (TEU) and forty-foot-equivalent-unit (FEU).

A CT main process life cycle starts after the arrival of vessels at the port, then the allocating of vessels to berths equipped with quay cranes (QCs). The QCs are responsible for loading and unloading the containers from and to the vessel. The QCs are equipped with trolleys that can move along the crane arm to transport the container from the ship to the transporting vehicle and vice versa. In most layouts, QCs are lined up alongside the quay, this setup makes it easy for the QCs to move along the quay, however, in this layout QCs can not pass each other. These containers are then transferred by trucks or trains to the yard area where they are temporarily stored until either transported inland or transferred to other vessels as shown by Figure 1. More details of port layout considerations are given by [3].

The seaside operation area in CT has three main sequential problems: Berth Allocation Problem (BAP), Quay Crane Assignment Problem (QCAP), and Quay Crane Scheduling Problem (QCSP). These problems could be solved independently, but this could be accepted for small instances (e.g small number of ships with few constraints) as it may only have a small negative impact on the performance. However, this is not the case for highly constrained large instances. [4] show that treating BAP and QCAP as an integrated problem could improve the overall performance by 34%. This paper address the first two problems as one integrated problem which is denoted as (BQCAP). The aim in BQCAP is to allocate each vessel of the incoming vessels with a quay space and starting time taking into consideration the availability of the number of QCs in the berth to assign them to the vessel.

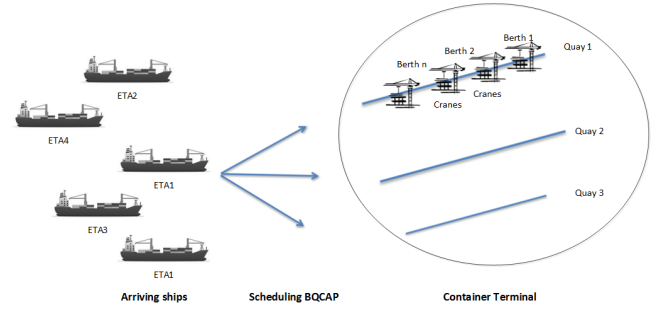


Fig. 2: The BQCAP problem description

B. Modelling the BQCAP

In order to solve the integrated BQCAP, we have a list of incoming vessels that needs to be served, the first part of the problem is to allocate a time and a space on the quay for all vessels BAP. Once a berth occupied for a vessel, no other vessel can occupy the same berth in the same time. The second part of the joint problem is to assign a set of QCs to serve the vessel all the duration of staying in the berth, taking into consideration that, the number of QCs should be within the minimum and maximum QCs required or contracted between terminal managers and ship owners in advance as shown by Figure 2. This number of QCs may decided depend on the length of a vessel, number of containers to be moved and how the containers distributed on a vessel. In this regard, vessel's operators should send the vessel information to the CT operators in advance. The information includes (e.g, vessel length, expected time of arrival, number of containers to be moved and its position on-board).

In this study, the BQCAP is modeled as a single objective function with the aim of minimizing the total cost of port stay times for all vessels. The problem definition and mathematical models obtained from [5] with minor modification.

Notation:

Input data and calculated variables:

- V : set of vessels to be served, $V = \{1, 2, \dots, n\}$.
- Q : number of available QCs.
- L : number of 10-m berth segments (length of the quay).
- T : set of 1-h periods, $T = \{0, 1, \dots, H - 1\}$, H is the planning horizon.
- l_i : length of vessel $i \in V$ given as a number of 10-m segments.
- b_i^0 : desired berthing position of vessel i .
- m_i : crane capacity demand of vessel i given as a number of QC-hours.
- r_i^{\min} : minimum number of QCs agreed to serve vessel i simultaneously.
- r_i^{\max} : maximum number of QCs allowed to serve vessel i simultaneously.
- R_i : feasible range of QCs assignable to vessel i , $R_i = [r_i^{\min}, r_i^{\max}]$.
- ETA_i : expected time of arrival of vessel i .
- EST_i : earliest starting time if journey of vessel i is speed up, $EST_i \leq ETA_i$.

EFT_i : expected finishing time of vessel i .
 LFT_i : latest finishing time of vessel i without penalty cost arising.
 c_i^1, c_i^2, c_i^3 : service cost rates for vessel i given in units of 1000 USD per hour.
 c^4 : operation cost rate given in units of 1000 USD per QC-hour.
 c_i^5 : possibility of overlap between vessel i and the rest of vessels.
 α : interference exponent.
 β : berth deviation factor.
 M : a large positive number.

Decision variables:

b_i : integer, berthing position of vessel i .
 s_i : integer, time of starting the handling of vessel i (berthing time).
 e_i : integer, time of ending the handling of vessel i (finishing time).
 r_{it} : binary, set to 1 if at least one QC is assigned to vessel i at time t , 0 otherwise.
 r_{itq} : binary, set to 1 if exactly q QCs are assigned to vessel i at time t , $q \in R_i$, 0 otherwise.
 Δb_i : integer, deviation between the desired and the actually chosen berthing position of vessel i , $\Delta b_i = |b_i^0 - b_i|$.
 ΔETA_i : integer, required speed up of vessel i to reach its berthing time, $\Delta ETA_i = (ETA_i - s_i)^+$.
 ΔEFT_i : integer, tardiness of vessel i , $\Delta EFT_i = (e_i - EFT_i)^+$.
 u_i : binary, set to 1 if the finishing time of vessel i exceeds LFT_i , 0 otherwise.
 y_{ij} : binary, set to 1 if vessel i is berthed below of vessel j , i.e. $b_i + l_i \leq b_j$, 0 otherwise.
 z_{ij} : binary, set to 1 if handling of vessel i ends not later than handling of vessel j starts, 0 otherwise.
 $v_i Order$: The order of vessel i in the V .

Mathematical Model:

$$\text{minimize } Z \quad (1)$$

$$= \sum_{i \in V} \left(c_i^1 \cdot \Delta ETA_i + c_i^2 \cdot \Delta EFT_i + c_i^3 \cdot u_i + c^4 \cdot \sum_{t \in T} \sum_{q \in R_i} q \cdot r_{itq} + c_i^5 \right) \quad (2)$$

$$d_i^{min} = \left\lceil \frac{(1 + \beta \cdot \Delta b_i) \cdot m_i}{(r_i^{max})^\alpha} \right\rceil \quad (3)$$

$$\sum_{t \in T} \sum_{q \in R_i} q^\infty \cdot r_{itq} \geq (1 + \beta \cdot \Delta b_i) \cdot m_i \quad \forall i \in V \quad (4)$$

$$\sum_{i \in V} \sum_{q \in R_i} q \cdot r_{itq} \leq Q \quad \forall t \in T \quad (5)$$

$$\sum_{q \in R_i} r_{itq} = r_{it} \quad \forall i \in V \quad \forall t \in T \quad (6)$$

$$\sum_{t \in T} r_{it} = e_i - s_i \quad \forall i \in V \quad (7)$$

$$(t+1) \cdot r_{it} \leq e_i \quad \forall i \in V \quad \forall t \in T \quad (8)$$

$$t \cdot r_{it} + H \cdot (1 - r_{it}) \geq s_i \quad \forall i \in V \quad \forall t \in T \quad (9)$$

$$\Delta b_i \geq b_i - b_i^0 \quad \forall i \in V \quad (10)$$

$$\Delta b_i \geq b_i^0 - b_i \quad \forall i \in V \quad (11)$$

$$\Delta ETA_i \geq ETA_i - s_i \quad \forall i \in V \quad (12)$$

$$\Delta EFT_i \geq e_i - EFT_i \quad \forall i \in V \quad (13)$$

$$M \cdot u_i \geq e_i - LFT_i \quad \forall i \in V \quad (14)$$

$$b_j + M \cdot (1 - y_{ij}) \geq b_i + l_i \quad \forall i, j \in V, \quad i \neq j \quad (15)$$

$$s_j + M \cdot (1 - z_{ij}) \geq e_i \quad \forall i, j \in V, \quad i \neq j \quad (16)$$

$$y_{ij} + y_{ji} + z_{ij} + z_{ji} \geq 1 \quad \forall i, j \in V, \quad i \neq j \quad (17)$$

$$s_i, e_i \in \{EST_i, \dots, H\} \quad \forall i \in V \quad (18)$$

$$b_i \in \{0, 1, \dots, L - l_i\} \quad \forall i \in V \quad (19)$$

$$\Delta ETA_i, \Delta EFT_i \geq 0 \quad \forall i \in V \quad (20)$$

$$r_{itq}, r_{it}, u_i, y_{ij}, z_{ij} \in \{0, 1\} \quad \forall i, j \in V \quad \forall t \in T \quad \forall q \in R_i \quad (20)$$

C. Model assumptions

The proposed model is established based on the following assumptions:

- The vessel can be moored on the quay if there is an empty space greater than or equal to vessel length and there is at least r_i^{min} available to start with.
- The number of QCs assigned to vessel i is R_i .
- The QCs can be assigned to vessels using a dynamic approach (time-variant), so that the QCs can be assigned to another vessel before the original vessel departs. For more details on the differences between static and dynamic assigning QCs see [6].
- Every vessel has a draft that is lower than the draft of the quay.
- The time of QCs movements along the quay as well as berthing and departure times of vessels to berth are not considered since it supposes a constant penalty time for all vessels.
- The quay is *Continuous layout*, the *ETA* is *Dynamic arrival*, the handling time of a vessel depends on the number of assigned QCs (*QCAP*) and the moves required also the *vessel position*, the *performance measure* is the total cost of (*position, waiting time,*

tardiness and overlap), which is the objective function is to minimize these costs. For more details on the problem classification see [7].

- The vessel length includes the required safety margins, which is the safe distance between two moored vessels.

III. LITERATURE REVIEW

A comprehensive survey of BAP and QCAP is given in [4]; moreover, it is continued in [8], who provide a review and classification for the journal papers published from 2004 to 2012 on container terminal seaside operations using the keywords: container, container terminal and port, filtering them to the ones solving the BAP, QCAP, and QCSP. The paper also provides good future trends for further researchers to improve seaside operation problems, including seaside layout and material handling equipment. A follow up survey to [4] is presented in [7]. The authors continued their work by classifying the new literature from 2009 to 2014 according to the future of models considered for berth allocation, quay crane scheduling and integrated approaches by using similar classification schemes to those proposed in [4]. The increase of BQCAP research indicates its growing importance within the field of container terminal operations optimisation.

[9] solve the integrated problem in another way. Since the container terminals process starts with BAP, then QCAP, then QCSP, the problem can be solved from back (QCSP) to forward (BAP & QCAP), taking the productivity of the available cranes as input data and constraints to solve the BAP and QCAP. The output of the BAP and QCAP is then used as input data to resolve and adapt the final QCSP. There are several objective functions defined in the literature for the BQCAP. The main objective function is to minimize the total service time for incoming vessels. In fact, real world container terminal operations are an uncertain system, due to the changeable weather conditions, breakdowns in QCs, changes of expected time of arrivals of the vessels, etc. Models and algorithms might be adapted to consider these unexpected situations since they cannot be predicted in advance. [10] formulate two conflicting objectives to solve the problem, which are minimizing the total service time and maximizing the robustness buffer time to accept uncertain situations. Their problem was solved by the Mixed Integer Linear programming (MILP) model to minimise the service time, and by a Genetic Algorithm (GA) model to maximize the robustness.

Priority rule or dispatching rule (DR) is the way how to tackle the order of vessel's list in order to schedule them. In literature, most of the researchers use first-come-first-serve (FCFS) rule to order Vessels depending on its ETA. Table II shows that less researches consider the DR, while some of them use DR as indirect in their solutions and they noted as priority-based rules to solve the seaside CT problems. The priority-based rules is either giving a fixed value and assigning it to vessel and that depending on many strategies and policies related to the terminal managers, or they assign a priority to vessel as variable in the case of congestion of the berth.

[6], applied three different DR rather than FCFS. (1) They use First Come Maximum Priority (FCMP): it is similar to FCFS where is the next vessel is chosen according to

their arrival order but, in this case, there is no restriction with the time the vessels can moor. (2) Maximum Weighted Waiting Time (MWWT): vessel's list ordered according to their weighted waiting time. The vessel with the highest value is moored first. (3) Earliest Weighted Mooring Time (EWMT): Among the vessels that can moor earlier, the operator chooses the vessel with the highest priority. [11], applied three rules, FCFS, giving priority to small vessels and giving priority to big vessels. [12], proposed simulation model that used two selection approaches, FCFS and Earliest Due Date (EDD) rule. [13], proposed a weighted shortest processing time first (WSPT) rule as a selection priority for incoming vessels. [5], propose two meta-heuristic approaches, which enable changes of the priority list in order to improve the quality of berth plans. [14], Applied a criterion to give a higher priority to large vessel with later EFT. [15], [16] and [17] used random priority rules to order vessel's list. [18], they used 26 priority rules randomly generated as an initial population using genetic algorithm. The first group of rules corresponds to individual characteristics of the vessels, while a second group combines these features into more complex rules. The rest of researches shown in Table II ignored FCFS rule and applied priority list for vessels. In this paper we use Genetic Programming to automatically evolve effective and robust composite dispatching rules (CDRs) to order the vessel's list.

TABLE II: Overview of researches applied Dispatching Rule in seaside CT operations

Year	Reference
2015	[19]
2015	[18]
2015	[20]
2014	[21]
2014	[6]
2014	[11]
2013	[14]
2013	[22]
2012	[13]
2012	[23]
2012	[24]
2012	[25]
2011	[12]
2011	[16]
2011	[17]
2009	[15]
2009	[5]
2008	[26]
2006	[27]
2003	[28]

IV. GENETIC PROGRAMMING FOR SOLVING BQCAP (GPBQCAP)

The proposed algorithm GPBQCAP consists of two main parts. The first part is the *Scheduler*, which is responsible for finding the best mooring time, berth location and assigning QCs for a given vessel v_i . The second part is the GP which responsible for discovering the best dispatching rule for a given port scenario. Flowchart Figure 4 describes the relationship between the *Scheduler* and the GP engine.

A. The Scheduler

To solve the BQCAP we represent the problem as a two-dimensional array, the x-axis demonstrates the time horizon,

and the y-axis demonstrates the quay length as shown by Figure 3. The goal is to plot all incoming vessels V in the array without violating the constraints mentioned earlier. A software program using java is developed to handle the plotting process for the vessel list one-by-one in the array, which will describe its function as (*Scheduler*). The order of the vessels are determined by the GP, which is described in the next subsection.

The scheduler receives the vessel's list ordered from the GP Engine and read vessel's data one by on.

The model starts by reading the vessel data and initialize the solution by: assigning Z to infinity, reset Q , filling the schedule array with zeros, calculate the minimum handling time for each vessel using equation (2) and check possibility of overlapping between incoming vessels and assign a cost c^5 for each vessel. We can calculate c^5 as it is the total area of intersection between two vessels overlapping. First, we check the availability of QCs in the time between vessels ETA and EFT . Second, we check if the desired position for the vessel is empty. If the QCs not available or overlapping with other vessel, we start testing all the times from EST to the time horizon H and all the possibility berths in the quay, then calculate the total cost of the vessel in this time and position using equation (1). Finally, we return the best time and position with minimum cost for the vessel. These processes continue till the end of vessel's list (V) and calculate the total cost of the schedule then send it to the GP engine. To speed up the running time we did stopping conditions if there is no improvement of the solution or no availability of QCs to the vessel, it returns the best solution found.

After the *Scheduler* places a vessel in the array, it fills the location with the correspondent Vessel ID. This shows the location of the vessel in the schedule. Empty spaces are denoted by zeros. A rectangle demonstrates the vessel with its position on the horizon starting from (s) and ending in (e) for the time horizon (x-axis), and the length of the rectangle shows the length of the vessel vl on the quay (y-axis).

We can notice that the best solution with total minimum cost can be obtained if all vessels can berth in desired berth position and ETA. If this is not the case there are two possible scenarios that can the solution be found and worse as follows:

- 1) If available QCs between vessel ETA and vessel $EFT < r^{min}$ for the vessel which leads the vessel to find another suitable starting time between EST and the time horizon.
- 2) If the desired position for a vessel at its ETA is occupied, this means that the vessel will overlap another vessel and that will leads the new vessel to shift from the desired position searching for a free space in the quay.

From equation (1), which calculate the total cost we can notice that scenario (1) is worse than the scenario (2), because shifting vessel from desired position will cost less than changing its starting time, but scenario (2) can be worse than scenario (1) if the distance of shifting exceed the limit and the vessel handling time duration begin to increase and it needs more quay cranes/hours to finish its loading or unloading required. Therefore, the proposed approach (*Scheduler*) can manage above scenarios by the following:

- 1) The *scheduler* starts to test every vessel with the rest of incoming vessels in the list if it is *Overlap Time* with it, if yes the *scheduler* will decrease the number of maximum quay cranes needed for all vessels that *Overlap Time* such that the total number of quay cranes for all vessels *Overlap Time* less or equal to the total number of QCS in the quay. And this will solve the above scenario (1).
- 2) The *scheduler* starts to test every vessel if the rest of incoming vessels in the list if it is *Overlap Position* with it, if yes the *scheduler* will start to shift the current vessel to be scheduled by adding coast c^5 which will allow shift the vessel on the quay from its desired position but not exceed the limit as described before. And this will manage the above scenario (2).

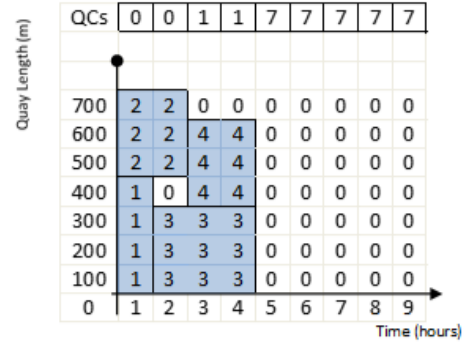


Fig. 3: Two-Dimensional array representation for the BQCAP

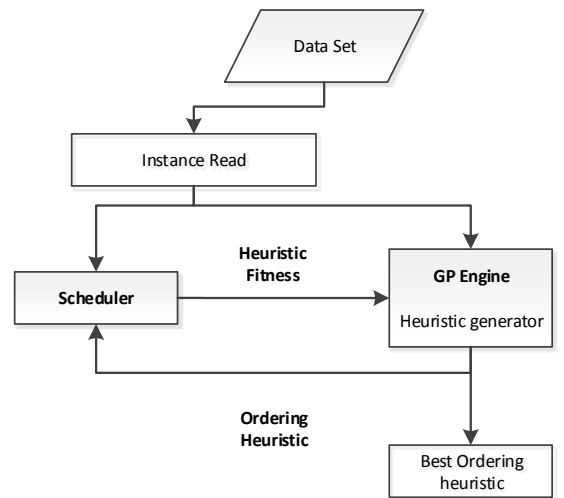


Fig. 4: GP Engine and *Scheduler* relationship

B. GP design framework

This paper presents a GP framework for evolving effective and robust composite dispatching rules (CDR) for solving the BQCAP. The GP Engine is responsible for ordering the vessel's list using CDR and send it to the *Scheduler* for testing the solution and calculating the fitness value from this order, then the GP engine try to evolve the CDR and testing it again in order to improve the quality of the generated solution as shown by Figure 4.

1) *Individual Structure*: Th paper adopts the standard GP tree presentation. Each individual presents a dispatching rule/equation, the equation is then evaluated for each vessel in the BQCAP instances, the results of the dispatching rule is consider the dispatch order of each vessel.

Each individual is constructed using the terminal and function sets. The function set consists of a number of mathematical operators as shown in Table III.

Each GP individual is represented as a tree based on a terminal set and a function set. In our model the initial individual is randomly generated. Figure 5 gives an example structure of the generated tree by GP that gives a possible CDR.

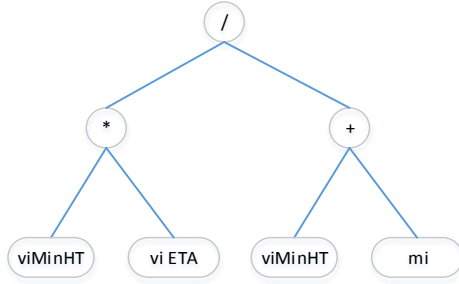


Fig. 5: Example of a GP tree with defined functions and terminals

2) *Terminal set*: The terminal set (leaf nodes) will be chosen as shown in Table III, these values can be found and calculated in the initialisation process as described before.

3) *Fitness function*: The obtained results from each generation of GP are a set of computer programs represented as a trees, the objective in our study is to find effective CDRs for solving the BQCAP with minimum total cost Z for all vessels V , shown in equation (1). Therefore, we propose a method to form a CDR from the tree-based result of GP. This CDR is then with the minimum cost evaluating the fitness value of the BQCAP. The following formula (21) represents a CDR (determines the order of a given vessel i) generated by the GP tree in Figure 5.

$$Order_Vessel(i) = \frac{v_i MinHT * v_i ETA}{v_i MinHT + m_i} \quad (21)$$

V. COMPUTATION AND EXPERIMENTS

The experiments are performed using the same benchmark introduced in [5]. The benchmark contains 20, 30 and 40

TABLE III: Function and Terminal sets.

Name	Description
Function set	
$Add(x, y)$	Addition
$Sub(x, y)$	Subtraction
$Mul(x, y)$	Multiplication
$Div(x, y)$	Safe division - returns 0 if the denominator equals to 0
$Avg(x, y)$	return the average value of the input variables.
$Min(x, y)$	Returns the minimum of the two inputs.
$Max(x, y)$	Returns the maximum of the two inputs.
$Abs(x)$	Return the absolute value of variable x .
$Ceiling(x)$	Return the ceiling value of the input.
$Floor(x)$	Return the floor value of variable x .
Terminal set	
vl	Vessel length
ETA_i	Vessel expected time of arrival for the given vessel i
ETF_i	Vessel expected time to finish for the given vessel i
EST_i	Earliest arrival time
LFT_i	Latest finish time
q_i^{Max}	The maximum number of QCs that can be assigned to a vessel
m_i	Number of movements (loading/unloading)
$vMinHT$	Time for handling a vessel while it is working with q_i^{Max}
v_{pos}	vessel's preferred mooring position.
pos	Preferred position.
$Overlap$	The amount of overlap with other vessels
$Tolerance$	Reflects the amount of time a vessel can wait and still finishes
$Density$	Amount of movement need per hour so the ship service can finish before the $vETA$
Gap	The length of unusable of the berth if allocated in its preferred position.
$Allowance$	The time between estimated time of finish and latest finish time.
$SpeedUp$	Duration between earliest time of arrival and expected time of arrival.
$Space$	The minimum required slots to serve the ship which is length of the ship multiplied by the minimum service time
$Volume$	The space multiplied by maximum number of QC.

vessels with ten instances each. The data for each vessel includes: vessel ID, length of vessel, desired berthing position, crane capacity demand of vessel, expected time of arrival, earliest starting time, expected finishing time, latest finishing time, minimum and maximum number of cranes to assign and vessel services costs. In this study, the following assumptions are considered:

- Quay length $L = 100(1000)$ metres
- Number of QCs on the quay $Q = 10 QC_s$
- Time Horizon $H = 168$ hour
- Interference exponent $\alpha = 0.9$
- Berth deviation factor $\beta = 0.01$
- Operation cost per QC-hour $c^4 = 0.1$

From the above assumptions and the input data, and using equation (2) we can initially calculate the best values for the handling time for the vessel $vMinHT$. In this experiment, the GP parameters are shown in Table V. GP is compared to the following standard priority rules (SPRs):

- *R1: FCFS*: The vessel list will ordered by ETA.
- *R2: Max QC needed low priority*: The vessel list will be ordered by r_i^{Max} Ascending.
- *R3: Max. QC needed high priority*: in this dispatching rule the vessel list will be ordered by r_i^{Max} descending.

- *R4: Vessel length high priority*: give the bigger vessels high priority in the sorting order.
- *R5: Vessel length low priority*: give the smaller vessels high priority in the sorting order.
- *R6: Min. Handling time low priority*: in this dispatching rule the vessel list will be ordered by $vMinHT$ ascending.
- *R7: Min. Handling time high priority*: same as above but gives the priority to vessels which need a long handling time.

In this first experiment, GP is compared to the SPRs explained above. Table IV shows the results between using SPRs and GP approach to solve BQCAP. The GP results performs better than the best results of SPR. *GP1* refer to GP with population size = 50.

In the second set of experiments, we used *GP2* with population size = 100 and we got better results than *GP1* but it takes longer time to obtain results. Up to our knowledge this is the only dataset available with existing results. Therefore, this dataset is used here as a benchmark to compare the GP results with other existing methods in the literature. Table VI shows the performance of the GP against other improvement and construction methods reported in [5]. FCFSLR uses the DR with local search, SWO uses the same DR with Squeaky wheel optimisation, while TS is based on Tabu Search optimisation method. The GP outperformed the FCFS construction approach. As expected the methods that uses both "construction and improvement" metaheuristics/heuristics has out performed the GP as a construction approach. However, the GP results are very competitive with the improvement methods. The aim in the future, is to combine the GP evolved heuristics with an improvement optimisation layer.

VI. CONCLUSION

This paper presents a new optimization method for the integrated berth allocation and quay crane assignment problem BQCAP, which aims to minimize the total costs of terminal services and operation for all incoming vessels. Moreover, the paper provides a review of the state-of-the-art of BQCAP methods that uses dispatching rules in their approaches.

In this paper we have introduced a genetic programming approach for automatically evolving efficient and customized dispatching rules for the berth allocation and the quay crane assigning problem GPBQCAP. In order to evolve efficient and customized DR, unlike standard dispatching rules, the GPBQCAP is capable of combining different sets of attributes (vessel related attribute set, berth related attribute set and quay crane related attribute set). This allows the GPBQCAP to generate tailored DR based on the attributes given in each set, as the attributes may vary from port to another. Results show that the GP outperform all SDRs in all instances (constructive) method and very competitive with the improvement methods.

For the further research, the aim is to extend the GPBQCAP to handle more complex quays layouts e.g. multiple quays layout, dual cycle cranes etc. Moreover, the aim is to test and evaluate the behaviour of the evolved heuristics when combined with other optimization methods. Also, the aim in

the future, is to combine the GP evolved heuristics with an improvement optimisation layer.

REFERENCES

- [1] UNCTAD, "Review of maritime transport 2015," *United Nations*, 2015.
- [2] S. Nguyen, M. Zhang, M. Johnston, and K. Chen Tan, "Hybrid evolutionary computation methods for quay crane scheduling problems," *Computers and Operations Research*, vol. 40, no. 8, pp. 2083–2093, aug 2013.
- [3] I. F. a. Vis and R. De Koster, "Transshipment of containers at a container terminal: An overview," *European Journal of Operational Research*, vol. 147, no. 2003, pp. 1–16, 2003.
- [4] C. Bierwirth and F. Meisel, "A survey of berth allocation and quay crane scheduling problems in container terminals," *European Journal of Operational Research*, vol. 202, no. 3, pp. 615–627, may 2010.
- [5] F. Meisel and C. Bierwirth, "Heuristics for the integration of crane productivity in the berth allocation problem," *Transportation Research Part E: Logistics and Transportation Review*, vol. 45, no. 1, pp. 196–209, jan 2009.
- [6] M. Rodriguez-Molins, M. a. Salido, and F. Barber, "A GRASP-based metaheuristic for the Berth Allocation Problem and the Quay Crane Assignment Problem by managing vessel cargo holds," *Applied Intelligence*, vol. 40, no. 2, pp. 273–290, aug 2014.
- [7] C. Bierwirth and F. Meisel, "A follow-up survey of berth allocation and quay crane scheduling problems in container terminals," *European Journal of Operational Research*, dec 2014.
- [8] H. J. Carlo, I. F. a. Vis, and K. J. Roodbergen, "Seaside operations in container terminals: literature overview, trends, and research directions," *Flexible Services and Manufacturing Journal*, pp. 1–39, jun 2013.
- [9] F. Meisel and C. Bierwirth, "A Framework for Integrated Berth Allocation and Crane Operations Planning in Seaport Container Terminals," *Transportation Science*, vol. 47, no. 2, pp. 131–147, may 2012.
- [10] M. Rodriguez-Molins, L. Ingolotti, F. Barber, M. a. Salido, M. R. Sierra, and J. Puente, "A genetic algorithm for robust berth allocation and quay crane assignment," *Progress in Artificial Intelligence*, apr 2014.
- [11] M. Ji, H. Zhu, Q. Wang, R. Zhao, and Y. Yang, "Integrated strategy for berth allocation and crane assignment on a continuous berth using Monte Carlo simulation," *Simulation*, vol. 91, pp. 26–42, nov 2014.
- [12] J. W. Böse, "Simulation of Container Ship Arrivals and Quay Occupation," in *Operations Research/ Computer Science Interfaces Series*, 2011, vol. 49, no. 1.
- [13] D. Xu, C. L. Li, and J. Y. T. Leung, "Berth allocation with time-dependent physical limitations on vessels," *European Journal of Operational Research*, vol. 216, no. 1, pp. 47–56, jan 2012.
- [14] M. H. Elwany, I. Ali, and Y. Abouelseoud, "A heuristics-based solution to the continuous berth allocation and crane assignment problem," *Alexandria Engineering Journal*, vol. 52, no. 4, pp. 671–677, dec 2013.
- [15] C. Liang, Y. Huang, and Y. Yang, "A quay crane dynamic scheduling problem by hybrid evolutionary algorithm for berth allocation planning," *Computers and Industrial Engineering*, vol. 56, no. 3, pp. 1021–1028, 2009.
- [16] C. Liang, J. Guo, and Y. Yang, "Multi-objective hybrid genetic algorithm for quay crane dynamic assignment in berth allocation planning," *Journal of Intelligent Manufacturing*, vol. 22, no. 3, pp. 471–479, jun 2011.
- [17] I. Ali, Y. Abouelseoud, and M. HamdyElwany, "Container terminal berth allocation and quay crane assignment using IP and simulated annealing," *41st International Conference on Computers Industrial Engineering*, pp. 31–37, 2011.
- [18] P. Frojan, J. F. Correcher, R. Alvarez-Valdes, G. Koulouris, and J. M. Tamarit, "The continuous Berth Allocation Problem in a container terminal with multiple quays," *Expert Systems with Applications*, vol. 42, no. 21, pp. 7356–7366, 2015.
- [19] E. Ursavas, "Priority control of berth allocation problem in container terminals," *Annals of Operations Research*, 2015.
- [20] M. Li, J. Jin, and C. Lu, "Real time disruption recovery for integrated berth allocation and crane assignment in container terminals," *optimization*, 2015.

TABLE IV: Performance comparison between the evolved GP scheduling and the SPRs.

ships	ins.	R1	R2	R3	R4	R5	R6	R7	GPI
20	1	118.5	132.7	124.6	90.2	132.7	94.1	106.8	88.1
	2	60.1	62.5	55.1	56.1	61.4	58.6	58	54
	3	94.7	119.9	84.3	134.7	108.2	161.8	93.1	81.2
	4	96.4	150.3	128.6	126.8	150.4	148.4	121.1	92.4
	5	65.2	77.2	57.7	57.9	69.4	72.4	57.7	57.7
	6	57.6	73.8	57.6	59.7	73.8	76.9	58.5	57.6
	7	88.3	99.1	77.1	77.1	104.8	101.1	77.1	77.1
	8	78.9	92.5	77.9	102.2	95.7	95.7	77.9	66
	9	86.5	104.7	90.1	84.9	105.6	124.4	93.2	81.9
	10	115.5	176.5	134.5	116.4	191.1	162.9	115.6	104.5
30	11	209	389.9	213.5	221.6	399.4	392.4	267.2	186.7
	12	96.5	145.6	92.3	100.9	145.3	146	104.9	88.9
	13	134.7	166.5	118.7	124.5	166.6	165.6	111.8	111.8
	14	128.9	284.3	137.6	136.4	285.1	159.8	141.2	118.7
	15	200.3	307.5	178.2	175.5	492.2	332.2	184.6	172.1
	16	135.9	155.6	129.4	130.8	216.9	203	129.6	126.2
	17	139.8	182.6	120	111.4	178.4	176.4	113.3	111.4
	18	167.8	223.6	166.8	157.1	222.7	242.3	160.3	139.7
	19	260.3	516.4	232	210.9	492.5	501.5	238.6	189
	20	153.9	269.4	146.5	150.4	313.3	442	161	140.3
40	21	322.4	551.7	393.1	321	637.7	754.1	430.1	260.7
	22	272.5	559.2	257.2	262.2	393	747.4	278.6	224.5
	23	476.8	926.3	451.2	452.6	1266.4	1083.9	363.9	305.9
	24	439.8	829.1	439.9	438.1	868.5	790.1	438.9	414.8
	25	221.2	578.7	228.6	258.8	867.6	622.7	191.8	175.4
	26	420.8	1110.8	486	430.2	1182	1011.4	414.4	343.1
	27	325.8	714.6	369	281.7	718	816	293.3	281.7
	28	423.1	1596.8	534.5	694.8	1281.3	1048.8	523.4	419.8
	29	293.5	622.5	259.8	414	569.5	747.5	413.6	243.2
	30	387.8	715.6	331.1	542.8	766.1	610.2	466.7	331.1

TABLE V: GP parameter values

Parameter	Value
Number of iteration	50
Population size	50
Number of generation	50
Maximum depth for population tree	7
Crossover probability	0.8
Mutation probability	0.1
Recreation probability	0.1

TABLE VI: Performance comparison between the evolved GP scheduling and the other heuristics.

ships	ins.	Construction			Improvement		
		FCFS	GP1	GP2	FCFSLR	SWO	TS
20	1	118.5	88.1	84.3	86.1	85.1	85.1
	2	60.1	54	54.29	53.9	53.9	53.9
	3	97.6	81.2	79.69	87.3	77.4	77.4
	4	96.4	92.4	90.4	79.7	79.7	77.9
	5	73.1	57.7	58.09	56.8	56.8	56.8
	6	57.6	57.6	57.5	57.6	57.6	57.6
	7	93.3	77.1	69.5	69.9	68.9	68.9
	8	78.9	66	61.9	69.6	57.0	56.1
	9	96.4	81.9	78.4	76.3	75.9	75.5
	10	115.5	104.5	98.59	101.1	94.6	93.0
30	11	216.0	186.7	178.3	152.6	147.8	149.5
	12	96.7	88.9	86.8	86.4	83.3	82.5
	13	135.0	111.8	112.3	107.6	105.7	104.5
	14	144.5	118.7	107.69	113.2	105.8	113.2
	15	197.5	172.1	145.3	173.8	159.0	157.4
	16	137.7	126.2	120.4	127.2	118.5	119.5
	17	139.8	111.4	110.5	110.2	104.5	104.2
	18	167.8	139.7	130.3	131.4	125.5	131.2
	19	268.7	189	185.7	185.0	173.8	173.8
	20	184.7	140.3	136.7	140.5	135.2	138.3
40	21	317.0	260.7	245.9	261.3	215.0	226.7
	22	276.9	224.5	222.4	189.0	178.8	183.4
	23	550.4	305.9	301.1	325.7	273.9	264.3
	24	453.3	414.8	410.5	360.2	326.6	342.2
	25	239.1	175.4	165.5	162.0	155.1	154.8
	26	398.9	343.1	304.3	273.1	260.4	259.6
	27	354.6	281.7	255.3	233.0	200.8	215.8
	28	424.2	419.8	390.5	408.5	286.2	294.3
	29	334.2	243.2	259.8	268.4	219.4	223.4
	30	425.8	331.1	326.2	280.8	240.9	254.7

- [21] A. H. Basri and Z. M. Zainuddin, "Development of integrated model for continuous berth allocation problem and quay crane scheduling with non crossing constraint," *STATISTICS AND OPERATIONAL RESEARCH INTERNATIONAL CONFERENCE*, vol. 113, pp. 113–121, 2014.
- [22] C. Yang, S. Wang, and J. Zheng, "The allocation of berth and quay crane by using a particle swarm optimization technique," in *Lecture Notes in Electrical Engineering*, vol. 254 LNEE, 2013, pp. 779–787.
- [23] Y. Xu, Q. Chen, and X. Quan, "Robust berth scheduling with uncertain vessel delay and handling time," *Annals of Operations Research*, vol. 192, no. 1, pp. 123–140, jun 2012.
- [24] L. Zhen and D. F. Chang, "A bi-objective model for robust berth allocation scheduling," *Computers and Industrial Engineering*, vol. 63, no. 1, pp. 262–273, aug 2012.
- [25] X. Liang, W. Li, W. Zhao, and B. Li, "Multistage collaborative scheduling of berth and quay crane based on heuristic strategies and particle swarm optimization," in *Proceedings of the 2012 IEEE 16th International Conference on Computer Supported Cooperative Work in Design, CSCWD 2012*, 2012, pp. 913–918.
- [26] A. Imai, H. C. Chen, E. Nishimura, and S. Papadimitriou, "The simultaneous berth and quay crane allocation problem," *Transportation Research Part E: Logistics and Transportation Review*, vol. 44, no. 5, pp. 900–920, sep 2008.
- [27] Z. Pengfei, K. Haigui, and L. Li, "A dynamic berth allocation model based on stochastic consideration," in *Proceedings of the World Congress on Intelligent Control and Automation (WCICA)*, vol. 2, 2006, pp. 7297–7301.

- [28] A. Imai, E. Nishimura, and S. Papadimitriou, "Berth allocation with service priority," *Transportation Research Part B: Methodological*, vol. 37, no. 5, pp. 437–457, jun 2003.